ซอฟต์แวร์
วิทยา สงวนวรรณ

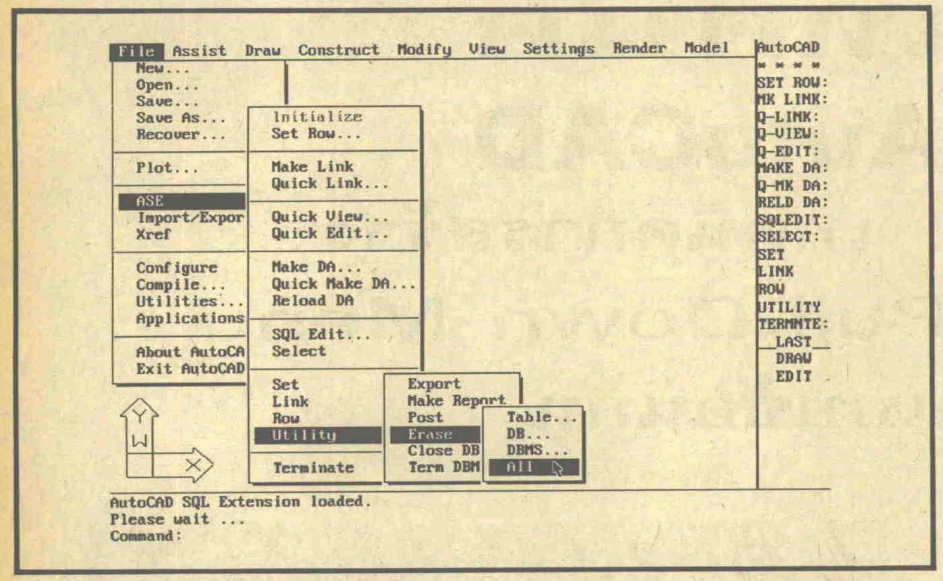
เทคนิค AutoCAD : เทคนิคการสร้าง Pull-Down Menu แบบซ้อนกัน

แนะเทคนิคและวิธีการสร้างเมนูแบบพลูดาวน์ที่ซ้อนกันอย่างง่าย ๆ เพื่อการใช้งานที่รวดเร็ว

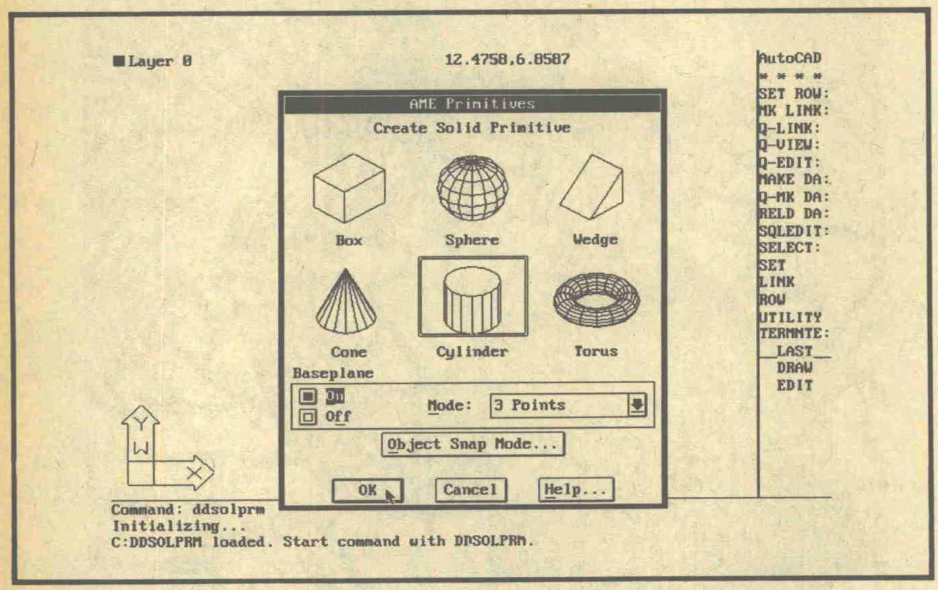
รีลีส 12 เปิดโอกาสให้ผู้ใช้สามารถออกแบบหน้าตา Pull-Down menu หรือที่บางท่านอาจจะเรียกว่า Popup menu ได้อย่างถึงใจมากยิ่งขึ้น จากเดิมที่เป็นเพียงพลูดาวน์ที่มีความสามารถเปิดเพียงคราวละ 1 วินโดว์เท่านั้น ซึ่งถ้าท่านรวบรวมคำสั่งกลุ่มใหญ่ ๆ สักกลุ่ม นำมาใส่ในเมนูตัวนี้ก็ดูเหมือนจะเป็นการบดบังรูปชิ้นงานไปเสียหมด Autodesk Inc. จึงได้มีการเปลี่ยนแปลงจากเดิมออกไป เปิดทางให้เรา ๆ ท่าน ๆ สามารถสร้างพลูดาวน์เมนูแบบ 3 มิติที่ซ้อนทับกันแบบสมบูรณ์

อะไรคือเมนูไฟล์

เมนูไฟล์ที่กล่าวถึงนี้ก็คือตัวโปรแกรมที่บรรจุมาโครและคำสั่งต่าง ๆ ที่ก่อกำเนิดเป็นรูปร่างหน้าตาของออโต้แคดเพื่อการอินเตอร์เฟสกับผู้ใช้นั่นเอง จะใช้ออโต้แคดง่ายหรือยากก็ขึ้นกับส่วนนี้แหละครับ ซึ่งท่านสามารถให้ออโต้แคดแปลงร่างเป็นอะไรก็ได้ ตั้งแต่งานด้านสถาปัตยกรรม โครงสร้าง เครื่องกล ไฟฟ้า ฯลฯ สุดแล้วแต่ท่านจะต้องการเลยทีเดียว ถ้ามาพิจารณาถึงส่วนต่าง ๆ ของการอินเตอร์เฟสกับผู้ใช้แล้ว บนเมนูนี้จะถูกแยกแบ่งออกเป็นส่วน ๆ อย่างชัดเจน เช่น ส่วนของสกรีนเมนู พลูดาวน์เมนู ไอคอนเมนู, Tablet menu ดังแสดงในรูปที่ 1 และ 2 ซึ่งซอร์สโปรแกรมนี้จะถูกเขียนขึ้นในรูปแบบแอสกีไฟล์ที่มีนามสกุล .MNU และท่านสามารถอ่านออกหรือแก้ไขได้อย่างง่าย ๆ



รูปที่ 1 แสดงพลูดาวน์เมนูและสกรีนเมนู



รูปที่ 2 แสดงไอคอนเมนู

จะแปลงร่างเป็นอะไรดี

ปกติการใช้งานของออโต้แคดในครั้งแรกหรือทั่ว ๆ ไปที่ผู้ใช้ยังไม่มีการปรับแก้ไขเมนูไฟล์จะถูกกำหนด (Default) ไว้กับไฟล์ที่ชื่อ ACAD.MNU เสมอ ตัวนี้แหละครับที่ท่านสามารถนำมาขัดเกลา ปรุงแต่ง จะให้ถูกอกถูกใจแค่ไหนย่อมกระทำได้อยู่แล้ว ซึ่งหน้าตาของออโต้แคดในคราวแรกที่เริ่มใช้งานออกจะไม่ทันสมัยเอามาก ๆ จะทำอะไรสักอย่างหนึ่งดูเหมือนจะต้องค้นคำสั่งอยู่ร่ำไป

สมมติว่าท่านเป็นผู้หนึ่งที่จะนำออโต้แคดมาดราฟต์ทางด้านโครงสร้าง แต่เมื่อนำออโต้แคดเข้ามาช่วยในงานออกแบบ อันดับแรกท่านจะต้องเสียเวลาส่วนหนึ่งทำความเข้าใจกับเมนูคำสั่งต่าง ๆ ของออโต้แคดที่ไม่ได้ใกล้เคียงกับภาษาทางโครงสร้างสักเท่าไร พอลงมือทำก็เริ่มจะพบว่าคำสั่งบางคำสั่งมันควรจะทำได้ง่ายกว่านี้ บางคำสั่งก็กลับดูเหมือนไม่ได้ถูกใช้เลย แต่บางคำสั่งกลับต้องใช้งานบ่อย ๆ แถมยังต้องเคาะแป้นอีกตั้งหลายทีเพื่อให้เข้าถึงคำสั่งนั้น หรือไม่ก็ต้องเปิดวินโดว์ซ้อนกันตัวที่ 2 ที่ 3 ให้ยุ่งยากเสียเวลามากเข้าไปอีก

เปลี่ยนแปลงเสียใหม่

ท่านก็จัดการปรุงแต่งหน้าตาของเมนูเหล่านี้ใหม่เสียสร้างเมนูเป็นของตัวท่านเอง จัดกลุ่มที่เป็นประเภทเดียวกันให้อยู่ด้วยกัน คำสั่งไหนที่ไม่ค่อยจะเรียกใช้หรือไม่ได้ใช้ก็ลบทิ้งไป คำสั่งไหนที่ต้องผ่านการเคาะแป้นคีย์บอร์ดตั้ง 3-4 ทีก็รวบจัดให้เหลือคำสั่งเดียว คำสั่งที่สร้างคาน เสา ประตู หลังคา ก็ใช้ AutoLISP, ADS เสริมเข้าไป เท่านี้ท่านก็จะได้เครื่องมือที่วิเศษ (Productive tools) เต็มประสิทธิภาพ 100 เปอร์เซ็นต์ ตรงกับเป้าหมายของท่านอย่างที่สุด

ซึ่งออโต้แคดเองก็ประกาศชัดแจ้งอยู่แล้วว่า ตัวมันเองเป็นสถาปัตยกรรมเปิด และก็ไม่ได้สร้างขึ้นเพื่อเน้นหนักทางด้านหนึ่งด้านใดโดยเฉพาะ ดังนั้นเมื่อนำมาใช้งานก็ดูเหมือนจะติดขัดอะไรบางอย่าง อย่างมากก็ใช้ได้แค่ 60-70 เปอร์เซ็นต์ จากที่ควรจะได้เท่านั้นท่านก็ต้องไปปรับแก้ (Customizing menu) เอาเองจึงจะได้ใช้แบบเต็มประสิทธิภาพ

ผ่า ACAD.MNU

ถ้าจับออโต้แคดเมนูทั้งหมดมาดูส่วนประกอบคร่าว ๆ ก็จะพบว่า ในกลุ่มของเมนูไฟล์นี้จะประกอบไปด้วยไฟล์ดังในรูปที่ 3 (อยู่ใน C:\ACAD\SUPPORT)

หน้าที่ของไฟล์แต่ละไฟล์พอสรุปได้ดังนี้

ACAD.MND

ACAD.MND จะเป็นไฟล์พิเศษที่กำหนดไฟล์ตัวอื่น ๆ (Menu definition) แต่จะใช้เฉพาะคราวแรกเท่านั้น ซึ่งจะต้องอาศัยการคอมพายล์จาก MC.EXE เพื่อสร้างซอร์สไฟล์ของ ACAD.MNU อีกทีหนึ่ง คล้าย ๆ กับเป็นต้นกำเนิดเพื่อสำรอง ACAD.MNU ไว้มากกว่าเผื่อถูกลบทิ้งหรือ Modify จะได้มีสำรองไว้

ACAD.MUL

ACAD.MUL จะเป็นไฟล์ของโปรแกรม AutoLISP (Menu lisp) ที่จะต้องมีการโหลดเข้าสู่หน่วยความจำอย่างอัตโนมัติทุกครั้งในขณะที่มีการเรียกใช้เมนู โดยจะต้องมีชื่อเดียวกับเมนูที่เรียกใช้งานด้วย แต่อย่างไรก็ตามจะมีหรือไม่มีก็ได้ ซึ่งหน้าที่จริง ๆ ของมันก็เพียงแต่ควบคุม ACAD.MNU ให้เป็นไปตามจุดประสงค์เท่านั้น

ACAD.MNU

ACAD.MNU จะเป็นตัวควบคุมส่วนประกอบต่าง ๆ ของรูปร่างหน้าตาในการอินเตอร์เฟสกับผู้ใช้ ซึ่งจะเป็นซอร์สโปรแกรมที่เป็นแอสกี สามารถอ่านออกหรือเข้าไปแก้ไขได้ และเมื่อใช้งานโปรแกรมออโต้แคดจะทำการคอมพายล์ให้กลายเป็น ACAD.MNX ที่ออโต้แคดนำไปใช้งาน

ACAD.MND	ขนาดประมาณ	90,059 bytes
ACAD.MNL	ขนาดประมาณ	9,054 bytes
ACAD.MNU	ขนาดประมาณ	95,739 bytes
ACAD.MNX	ขนาดประมาณ	153,203 bytes

รูปที่ 3 แสดงกลุ่มไฟล์ที่จำเป็นในการสร้างเมนูของออโต้แคด

ACAD.MNX

ACAD.MNX ไฟล์ตัวนี้จะเป็นไฟล์ที่ออโต้แคดนำไปใช้งานจริง ซึ่งจะได้จากการคอมพายล์ ACAD.MNU ให้กลายเป็นภาษาที่ออโต้แคดเข้าใจ แต่เราจะอ่านไม่รู้เรื่อง แต่ถ้าต้องการจะไปแก้ไขส่วนใดส่วนหนึ่งก็ต้องไปแก้ไขที่ตัว ACAD.MNU แทน แล้ว AutoCAD จะมีการคอมพายล์ใหม่อย่างอัตโนมัติ

แต่ถ้าเอาเฉพาะ ACAD.MNU มาพิจารณา โดยใช้โปรแกรมประเภทอีดิตเตอร์ทั่วไปเปิดดูก็จะพบว่าในซอร์สโปรแกรมจะถูกจัดและแบ่งเซกชันอย่างเป็นระบบ เป็นพื้นที่ของในแต่ละส่วนเมนูอย่างชัดเจน ซึ่งขอเรียกสั้น ๆ ว่า Menu Area ที่มีส่วนประกอบคือ

- Screen menu
- Pull-down และ Cursor menus
- Icon menu
- Pointing device buttons menus
- Digitizing Tablet menus
- Auxiliary device menus

ในแต่ละส่วนจะถูกแยกกันอยู่อย่างเป็นระบบ ไม่ปะปนกันในไฟล์ ACAD.MNU และในขณะที่ใช้งานในทุก ๆ ส่วนของเมนูแอเรียจะถูกโหลดเข้าสู่หน่วยความจำทั้งหมด ซึ่งก็ขึ้นอยู่กับตัวท่านเองว่าจะเลือกใช้ส่วนใดของเมนู ยกตัวอย่าง สมมติท่าน Configuration เพื่อการใช้เม้าส์ ก็จะได้ใช้ทุกส่วนยกเว้น Digitizing tablet menu เท่านั้นครับ

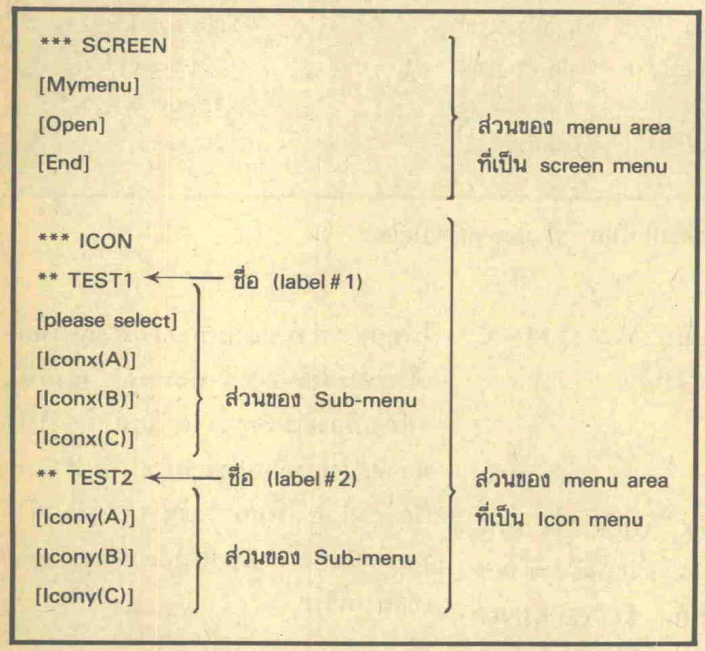
Menu Area ผู้สั่งการ

ในการสร้างเมนูขึ้นใช้เองจะต้องมีส่วนประกอบอย่างใดอย่างหนึ่ง ซึ่งไม่จำเป็นว่าจะต้องมีเมนูแอเรียครบทุกส่วนก็ได้ และในแต่ละส่วนจะประกอบไปด้วยส่วนย่อย ๆ ลงไปอีก ที่เรียกว่า Sub-menu ส่วนนี้แหละที่นำไปใช้งานหรือนำไปแสดงบนจอภาพ ที่สำคัญมากก็คือจะต้องกำหนดชื่อ (Label) ไว้ทุก ๆ เมนูแอเรียเสมอ เพื่อสะดวกในการถูกเรียกใช้งาน ดังเช่นในรูปที่ 4

ถ้าสังเกตให้ดีจะพบว่าโปรแกรมจะถูกแบ่งเป็นส่วน ๆ ส่วนที่ใหญ่สุดจะเป็นเมนูแอเรียที่มี "***" เป็นตัวแสดงถึงส่วนเริ่มต้นของเมนูแอเรีย และในแต่ละส่วนนี้ยังแบ่งย่อยเป็นซับเมนูที่มี "**" เป็นตัวแสดงถึงขอบเขตของแต่ละซับเมนู และที่เน้นก็คือจะต้องมีชื่อ (Label) กำกับไว้ทุก ๆ อันด้วย คล้าย ๆ ติดป้ายกันหลงนั้นเอง

เมนูอิสระ

ในส่วนของเมนูแอเรียนี้ใน ACAD.



รูปที่ 4 แสดงส่วนประกอบย่อยในเมนูแอเรีย

ตารางที่ 1 แสดงถึงตัวอักขระที่ควบคุมส่วนต่าง ๆ ของเมนูแอเรีย

Menu area	ส่วนควบคุม
*** BUTTONSn	ควบคุม Pointing device button menu n จะมีค่าตั้งแต่ 1 ถึง 4
*** AUXn	ควบคุม Auxiliary device menu n จะมีค่าตั้งแต่ 1 ถึง 4
*** POPn	ควบคุม Pull-Down/Cursor menu n จะมีค่าตั้งแต่ 0-6
*** ICON	ควบคุมไอคอนเมนู
*** SCREEN	ควบคุมสกรีนเมนู
*** TABLETn	ควบคุม Tablet menu n จะมีค่าตั้งแต่ 1 ถึง 4

MNU จะมีอยู่ด้วยกันอยู่ 6 ประเภท ด้วยกัน แยกเป็นอิสระไม่ปะปนกัน แต่ละส่วนก็คอยกำกับลูกทีมคือซับเมนูในส่วนของตัวเองไว้อย่างเป็นระบบ ซึ่งจะสรุปได้ดังในตารางที่ 1

ซึ่งแต่ละส่วนจะทำหน้าที่ควบคุมซับเมนูและมาโครที่อยู่ในแอเรียของตัวเองอย่างเป็นอิสระ ไม่ปะปนกัน แต่ถ้าหากท่านจะต้องการที่จะให้แต่ละเมนูแอเรียสามารถเรียกใช้ซับเมนูที่อยู่ต่างเมนูแอเรีย ก็สามารถกระทำได้โดยอาศัยการเรียกใช้ชื่อ (lable) ที่ท่านได้ตั้งไว้ในแต่ละส่วนของซับเมนูนั่นเอง ซึ่งลักษณะนี้จะเรียกว่า Referencing a Sub-menu ส่วนนี้แหละครับที่จะทำให้ผู้ใช้งานจะประทับใจหรือไม่ประทับใจก็อยู่ตรงนี้ รวมถึงการใช้งานที่ง่ายหรือยากด้วย

ในการใช้งานจริง ๆ ท่านสามารถกำหนดวิธีการอินเตอร์เฟสได้หลายรูปแบบ แต่แนวทางที่ดีที่สุดก็คือ ทำอย่างไรให้ใช้โปรแกรมออโต้แคดได้ง่ายที่สุด ซึ่งก็เป็นหน้าที่ของผู้พัฒนาที่จะต้องใช้การ Referencing a Sub-menu ให้เป็นประโยชน์มากที่สุดด้วย ยกตัวอย่างเช่น ท่านต้องการเขียนหน้าต่างดราฟต์บน Drawing ท่านก็เรียกคำสั่งวาดรูปหน้าต่างจากพลูดาวน์เมนู เมื่อต้องการให้รู้ว่าเป็นชนิดไหนก็ให้แสดงรูปร่างกราฟิกบนไอคอนเมนูแล้วไป Pick เลือกตามชนิดที่มองเห็นบนจอภาพ เมื่อได้ชนิดของหน้าต่างแล้วโปรแกรมก็ให้ป้อนความกว้างของหน้าต่าง ซึ่งก็ควรจะเลือกตัวเลขจากสกรีนเมนูด้านข้างจอได้เลย ก็เป็นอันว่าท่านก็สามารถดราฟต์หน้าต่างได้อย่างสบาย ๆ โดยแทบจะไม่แตะต้องคีย์บอร์ดเลยครับ

Referencing Sub-menu ผู้สร้างความแพรวพราว

การทำ Referencing Sub-menu จะช่วยเพิ่มเสน่ห์ให้กับออโต้แคดทำให้ดูเป็นระบบและน่าใช้มากยิ่งขึ้น เป็นการสร้างความสัมพันธ์ขึ้นมาในแต่ละซับเมนู ซึ่งวิธีการนี้จะทำให้การใช้งานมีประสิทธิภาพมากขึ้น แทนที่จะจำกัดการใช้งานอยู่เฉพาะแต่ละกลุ่มของเมนูแอเรียเท่านั้น รูปแบบการใช้งานคือ

```
$Section = Submenu
```

โดยที่

– **$** เป็นสัญลักษณ์นำหน้าตัวเลือกส่วนของเมนูแอเรีย

– **Section** จะเป็นแต่ส่วนของเมนูแอเรีย

s สำหรับสกรีนเมนู

$P_0 - P_{16}$ สำหรับ Pop menu ตั้งแต่ 0-16

I สำหรับไอคอนเมนู

$B_1 - B_4$ สำหรับบัททอนเมนูตั้งแต่ 1-4

$T_1 - T_4$ สำหรับ Table menu ตั้งแต่ 1-4

$A_1 - A_4$ สำหรับ Auxiliary menu ตั้งแต่ 1-4

– **Submenu** จะเป็นชื่อ (label) ที่เราตั้งชื่อใส่ไว้เพื่อเรียกใช้ได้ ยกตัว-

อย่างเช่น $S = X, $I = ICONX, $P = POP3, $I = PDMODE ฯลฯ แต่ในกรณีจะต้องการเรียกใช้ซับเมนูที่ผ่านมาครั้งล่าสุด (Previous) จะต้องใช้ $Section = แทน โดยทิ้งว่างไว้เฉย ๆ โปรแกรมก็จะเข้าใจว่าท่านกำลังเรียกใช้ซับเมนูอันเดิมที่เคยถูกเรียกใช้มาแล้ว ต่อไปเรามาลองดูตัวอย่างสักตัวอย่างหนึ่งที่มีการเรียกใช้ซับเมนูตัวอื่นโดยผ่าน Referencing submenu (ดูรูปที่ 5)

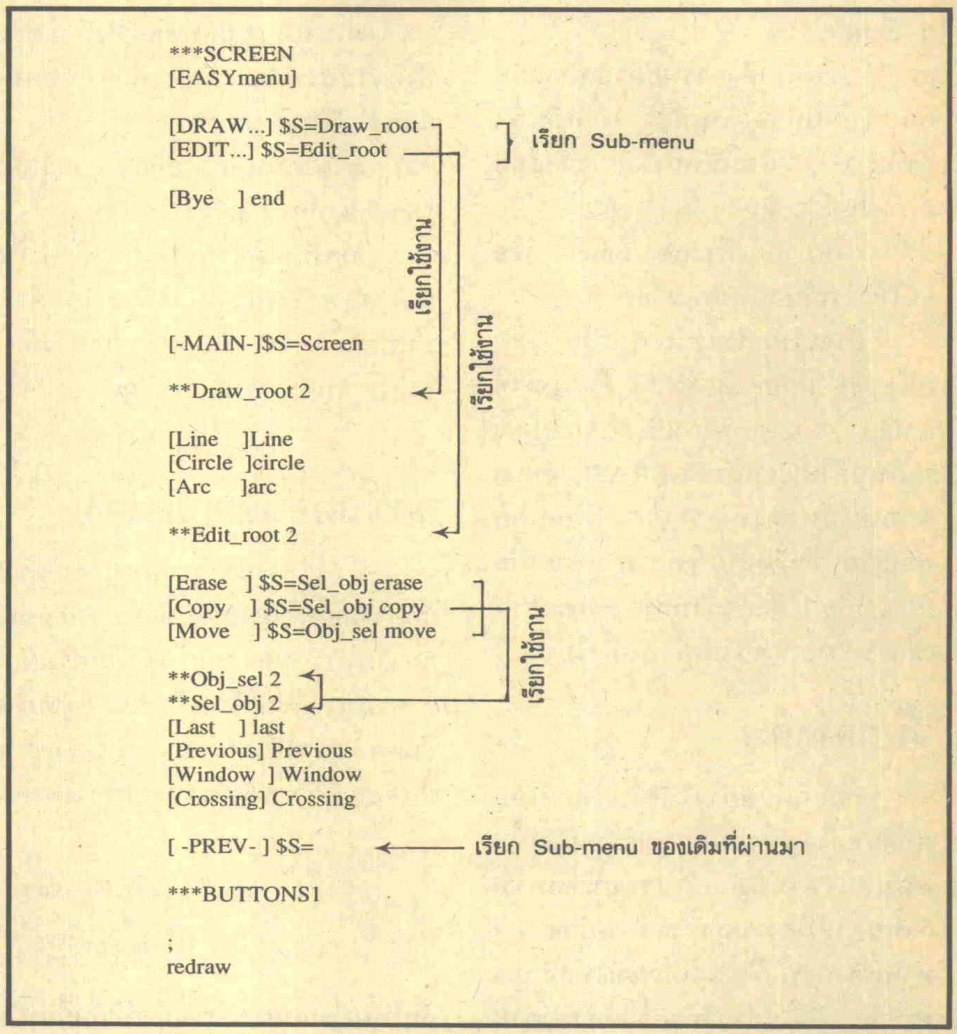
เมื่อพิมพ์เสร็จอย่าลืมจัดเก็บลงในซับไดเรคทอรีให้ถูกต้องด้วยนะครับ เดี๋ยวจะเรียกไม่ขึ้น สมมติว่าตั้งชื่อไว้ว่า "Refer.mnu" จากนั้นก็ให้พิมพ์คำสั่งเพื่อเรียกเมนูมาใช้งาน

```
Command : MENU
```

ก็จะเกิดไดอะล็อกบ็อกซ์ขึ้นมาใช้ไปเลือก Pick ที่ Refer.mnu แล้วทดลองใช้งาน ดูวิธีการทำงานที่เป็นแบบ Referencing Submenu จะเป็นลักษณะแบบใด โดยจะแสดงเฉพาะสกรีนเมนูเพียงอย่างเดียว ในตัวโปรแกรมจะพบว่ามีการใช้ $S = Draw_Root เพื่อเรียกซับเมนู "** Draw_Root" ขึ้นมาใช้งาน ซึ่งจะต้องมีชื่อเดียวกันเสมอนะครับ ส่วน "$S" นั้นจะหมายถึงเรียกใช้ส่วนของเมนูแอเรียที่เป็น screen แต่ถ้าต้องการเรียกใช้จากส่วนอื่นเช่นไอคอนเมนู ก็ใช้ $I หรือถ้าเป็นสกรีนเมนูก็ใช้ $S แทน

มาโครผู้สั่งการบนออโต้แคด

"มาโคร" ซึ่งท่านก็คงคุ้นหูกันดี ไม่ว่าจะเป็นมาโครของโลตัส มาโครของฟอกซ์โปร คราวนี้ก็คงมาถึงมาโครของออโต้แคดกันดูบ้าง โดยหลักการก็คงยังคล้าย ๆ กันอยู่ ถ้าเกิดถามว่ามาโครทำอะไรได้บ้าง ก็สรุปแบบสั้น ๆ ได้ว่าคือ *"การนำคำสั่งต่าง ๆ ของออโต้แคดมาเรียงกันตามลำดับ โดยมีตัวอักขระควบคุมการทำงานให้เป็นไปตามขั้นตอนที่ได้วางเอาไว้"* มีประโยชน์มากมายครับ ไม่ว่าจะเป็นการลดงานที่จะต้องทำซ้ำซ้อนกันเรื่อย ๆ หรือรวบรัดขั้นตอนที่เยิ่นเย้อให้เหลือเพียงแค่สั้น ๆ ถ้าเขียนมาโครดี ๆ ก็สามารถเขียนแบบแปลนตึกสวย ๆ ได้ เพียงแต่ป้อนจำนวนชั้นและความสูงของแต่ละชั้นเท่านั้นเอง ที่เหลือโปรแกรมเขียนอย่างอัตโนมัติหมดเลย ไม่ใช่เรื่องที่เกินความจริงแต่ประการใดเลย ต่อไปจะเป็นการเรียนลัดมาโครโดยขอยกตัวอย่างที่ตัดมาจากเมนูสักตัวอย่างหนึ่ง



รูปที่ 5 แสดงถึงความสัมพันธ์แบบ Referencing Sub-menu

```
[ERASE 1]^C^C ERASE; \ ;
```

[ERASE 1] จะเป็นการนำข้อความที่อยู่ในเครื่องหมาย [ ] ไปปรากฏบนเมนูในหน้าจอ ซึ่งจะแสดงข้อความเฉพาะที่อยู่ในก้ามปูเท่านั้น

^C^C จะหมายถึงการ ctrl+c ที่ยกเลิกคำสั่งเดิมให้กลับสู่คอมมานด์ไลน์ ในการเขียนให้กด shift ค้างไว้ ตามด้วย 6 ก็จะปรากฏ "^" และพิมพ์ C ก็จะได้ "^C"

ERASE เป็นคำสั่งออโต้แคดที่ถูกเรียกใช้งาน

\ หมายถึงการหยุดรอเพื่อเลือกออปเจ็กต์บนจอภาพเพียง 1 ครั้ง ซึ่งถ้ามี \ \ ก็จะเป็นการหยุดรอเลือกออปเจ็กต์ 2 ชิ้น

; หมายถึงการกด Enter หรือจะใช้การเว้นวรรคแทนก็ได้

ถ้าจะให้กล่าวรวม ๆ ก็คือ เมื่อ Pick ที่ตัวเลือก ERASE 1 ที่เมนู ก็ให้ยกเลิก ^C^C คำสั่งเก่าเสียด้วย ctrl+c จากนั้นก็ให้เรียกคำสั่ง ERASE ขึ้นมาพร้อมกับหยุดรอ "\" เพื่อเลือกออปเจ็กต์ที่จะลบบนจอภาพ 1 ชิ้น เมื่อ Pick เลือกได้แล้วก็ให้กด Enter ";" เพื่อรับชิ้นงานเข้าไปเพื่อลบทิ้งไป

ขั้นตอนเดิม

ขอยกตัวอย่างให้เห็นชัดเจนอีกสักตัวอย่างหนึ่งเพื่อความชัดเจนยิ่งขึ้น สมมติว่าท่านมีตัวอักขระบนจอภาพกำหนดให้มีความสูง 0.2 หน่วย แต่ท่านต้องการแก้ไขตัวเท็กซ์นี้ให้สูง 0.5 หน่วย ถ้าไม่มีมาโครก็จะต้องทำตามขั้นตอนดังนี้

- ใช้คำสั่ง CHANGE
- โปรแกรมจะให้ Select object: ให้เลือกตัวอักษรนั้นแล้วกด Enter #1
- รับค่าออปเจ็กต์เข้าไปด้วยการกด Enter #2
- โปรแกรมจะให้เลือกเปลี่ยน Properties/ <Change point> ต้องการรับค่า Default (ค่าเดิม) ด้วยการกด Enter #3
- โปรแกรมจะถามถึงตำแหน่งที่จะวางใหม่ ให้กด Enter #4 รับค่า Default
- โปรแกรมจะถามรูปแบบ Style ของตัวอักขระให้รับค่า Default ด้วย Enter #5
- ขั้นตอนนี้แหละครับที่จะถามถึงความสูงที่ต้องการจะเปลี่ยนป้อนค่า 0.5 เข้าไป
- จากนั้นก็กด Enter ผ่านรับค่า Default อีก 2 ครั้ง

จะเห็นได้ว่าการจะเข้าถึงคำสั่งและหลุดรอดออกมาได้จะต้องผ่านขั้นตอนถึง 9 ขั้นตอน และยังต้อง Enter รับค่า Default อีกตั้ง 7 ครั้ง

ลดขั้นตอนด้วยมาโคร

ซึ่งถ้าหากต้องเปลี่ยนเท็กซ์ทั้งข้อความและความสูงทั้งแผ่น Drawing ก็คงไม่น่าที่จะสนุกนัก ใช่ไหมครับ แต่ถ้าหากท่านใช้มาโครเข้ามาช่วยรวบรัดขั้นตอนเหล่านี้ให้เหลือแค่ความสูงเพียงขั้นตอนเดียวเสีย ด้วยการใช้มาโครคือ

```
[Change Height] *^C^C Change;
\;;;;\;;
```

ที่น่าสนใจอีกประการหนึ่งก็คือ เมื่อท่านได้เรียกให้ออโต้แคดทำงานตามมาโครของท่านที่ได้วางไว้แล้ว และยังต้องการที่จะวนลูปกลับมาใช้คำสั่งเดิมอีกไปเรื่อย ๆ จนกว่าจะกดคีย์ Ctrl+c ก็ให้ใช้ "*" นำหน้าคำสั่งบนเมนูนั้น เช่น

```
[Change Height]**^C^C Change; \;;;;\;;
```

แสดงการเรียกคำสั่งซ้ำอีก

เหตุผลที่จะต้องทำเช่นนี้เพราะการเปลี่ยนความสูงของเท็กซ์บน Drawing อาจจะไม่ได้กระทำเฉพาะเท็กซ์แค่ตัวเดียว แต่อาจจะต้องกระทำเป็นกลุ่มหลายตัว ถ้ามีการวนลูปเช่นนี้จะสะดวกในแง่ไม่ต้องกด Enter ทุกครั้งเพื่อเรียกคำสั่งเดิม

รายละเอียดเมนู

เพื่อความสะดวกต่อการใช้งานและการเขียนมาโคร ผมจึงได้สรุปรายละเอียดทั้งหมดของมาโครดังในตารางที่ 2 ดังนี้

เข้าใจพลูดาวน์เมนูและเคอร์เซอร์เมนู

ความจริงแล้วทั้งพลูดาวน์เมนูและเคอร์เซอร์เมนูต่างก็เป็น Popup menu ทั้งคู่ คือสามารถเกิดขึ้นมาลอย ๆ ได้เลย ซึ่งถ้าจะให้แยกข้อแตกต่างของทั้งสองก็จะอยู่ที่การควบคุมให้แสดงตัวมันเองออกมา พลูดาวน์เมนูจะเกิดตำแหน่งที่คงที่ และจะเกิดคู่กับแถบเมนูบาร์ที่อยู่บนสุดของจอภาพเสมอ แต่เคอร์เซอร์เมนูจะเกิดตำแหน่งไหนก็ได้บนจอภาพ และจะปรากฏแบบทันทีเลย โดยไม่ต้องรอให้เกิดแถบเมนูบาร์ขึ้นก่อน

ในการเขียนเมนูของทั้งสองจะมีข้อแตกต่างกันบ้างเล็กน้อย เคอร์เซอร์เมนูจะถูกกำหนดตายตัวไว้เฉพาะใน ***POP0 เท่านั้น ส่วนพลูดาวน์เมนูจะถูกกำหนดตั้งแต่ ***POP1 ไปถึง ***POP16 (มี 16 menu) ซึ่งการเขียนมาโครควบคุมก็เช่นเดียวกับที่ผ่านมา แต่จะมีข้อมูลปลีกย่อยเพิ่มขึ้นอีกเล็กน้อยคือ การเกิด Popup ในรีลีส 12 นี้จะสามารถซ้อนทับกันอย่างอัตโนมัติได้ และตัวจักรสำคัญที่ใช้ในการควบคุมการเกิดวินโดว์ซ้อนกันจะอยู่ที่สัญลักษณ์ตัวเดิมข้างหน้าที่อยู่ในเมนูนั้น

"→" จะแสดงถึงส่วนเริ่มต้นให้มีการเกิด Popup ตัวใหม่

"←" เป็นส่วนที่แสดงถึงคำสั่งสุดท้ายของ Popup ตัวนี้

ตารางที่ 2 สรุปตัวอักขระควบคุมเมนู

ตัวอักขระควบคุม	รายละเอียด
***	เป็นส่วนแสดงกลุ่มคำสั่งในเมนูแอเรีย
**	เป็นส่วนแสดงถึงกลุ่มคำสั่งในซับเมนู
[xxxx]	จะเป็นการนำข้อความ XXXX ที่อยู่ระหว่าง [ ] นำไปปรากฏการณ์เมนูต่าง ๆ เพื่อโต้ตอบกับผู้ใช้
;	มีค่าเท่ากับการกด Enter
^M	มีค่าเท่ากับการกด Enter เช่นกัน
^I	มีค่าเท่ากับ Tab
\	หยุดรับค่า 1 ครั้ง โดยโปรแกรมจะให้ Select object:
_	ทำหน้าที่แปลคำสั่งออโต้แคด
+	ใช้ในการต่อเมนูไปยังบรรทัดถัดไปเมื่อไม่สามารถเขียนจบบรรทัดเดียวได้
=*	ให้ทำการแสดงเคอร์เรนต์ไอคอน, พูลดาวน์ หรือเคอร์เซอร์เมนูบนจอภาพขณะนั้น
*^C^C	เมื่อต้องการทำชุดคำสั่งมาโครชุดนั้นซ้ำไปอีกเรื่อย ๆ
$	เป็นอักขระพิเศษเมื่อใช้เรียกซับเมนูมาใช้งาน เช่น $S, $I, $Po เป็นต้น
^B	ควบคุมการสแนปให้ปิด/เปิด
^C	ควบคุมการยกเลิกคำสั่งเดิมมีค่าเท่ากับการกด ctrl+c
^D	ควบคุมโคออร์ดิเนตให้ เปิด/ปิด
^E	ควบคุม Isometric plane
^G	ควบคุมการเกิด Grid ให้ เปิด/ปิด
^H	มีความหมายเท่ากับ backspace
^O	ควบคุมการทำ Otho ให้ เปิด/ปิด
^P	ควบคุม MENUECHO ให้ เปิด/ปิด
^Q	ควบคุมการนำข้อความจากเท็กซ์โหมดสู่พรินเตอร์
^T	ควบคุม Tablet ให้ ปิด/เปิด
^V	ควบคุมการแสดง Current Viewport

แยกเมนูไม่ให้ปนกัน

นอกจากนี้ยังมีตัวอักขระที่ใช้ในการควบคุมในการแบ่งแยกเมนูให้เป็นระบบไม่ปะปนกันคือ

[– –] จะใช้แยกส่วนของเมนูทำให้เกิดมีการขีดเส้นกันขึ้น

[~] จะทำให้เกิดช่องว่างขึ้นระหว่างพลูดาวน์เมนู

เฉพาะส่วนของเคอร์เซอร์เมนูนี้จะสามารถแสดงได้ด้วยการควบคุมของเม้าส์ ซึ่งเม้าส์นี้เองจะมีปุ่มควบคุมอยู่ 2-3 ปุ่มแล้วแต่ยี่ห้อที่ผู้ผลิตสร้างขึ้นมา ไม่ว่าจะเป็นทั้งแบบ 2 หรือ 3 ปุ่ม ก็จะใช้แบบเดียวกันคือ ปุ่มซ้ายจะเป็นปุ่มที่ใช้ในการเลือกออปเจ็กต์ไม่สามารถปรับแก้ไข ส่วนปุ่มทางขวาสุดจะเป็นปุ่ม Pick ที่เราสนใจจะสามารถปรับแก้ได้ ส่วนแบบ 3 ปุ่มจะมีปุ่มกลางที่เปิดโอกาสให้ผู้เขียนโปรแกรมสามารถเรียกใช้ได้เช่นกัน แต่จะไม่กล่าวถึงในที่นี้

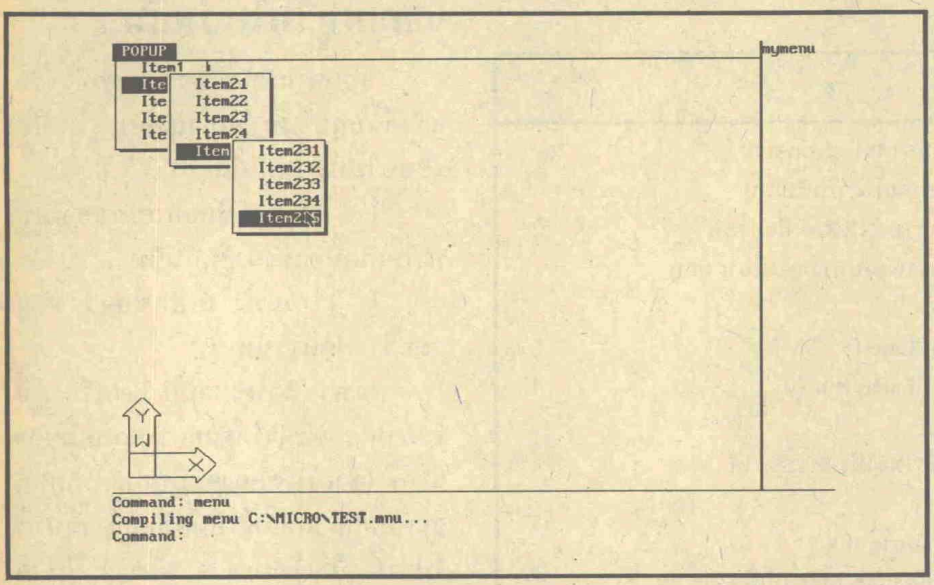
ในปุ่มทางขวาสุดจะใช้ควบคู่กับการกดคีย์บนแป้นพิมพ์ ซึ่งออโต้แคดได้แยกความสัมพันธ์ออกเป็นตารางดังในตารางที่ 3

จากตารางที่ 3 จะใช้การกดคีย์บนแป้นพิมพ์ค้างไว้พร้อมกับปุ่ม Pick ทางขวามือ ถ้าท่านจะต้องการให้แสดงเมนูสัมพันธ์กับปุ่มเหล่านี้ก็ต้องเป็นเขียนส่วนของเมนูให้สัมพันธ์กับบัททอนด้วย

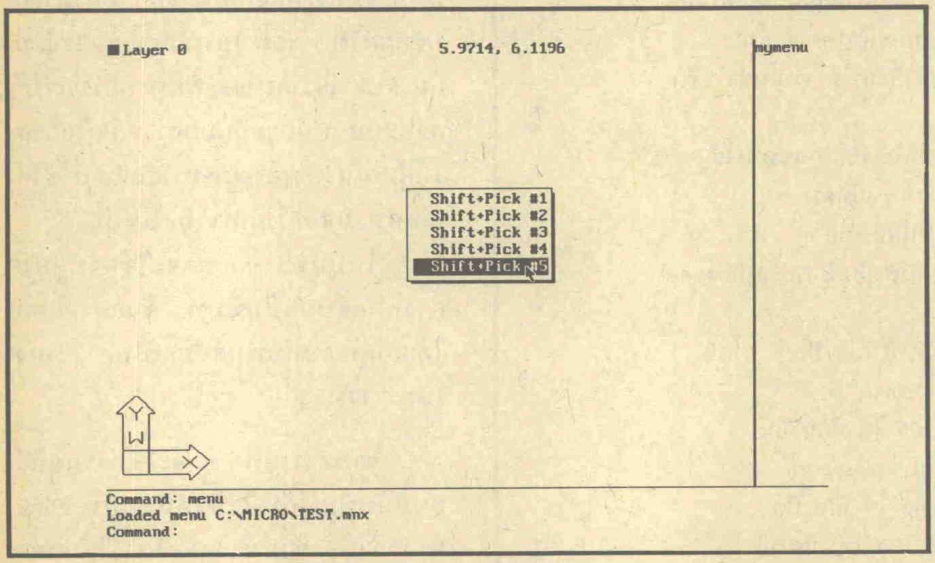
แต่ถ้าสมมติว่ามีการเกิดเมนูหลาย ๆ ตัว ก็ยังสามารถให้สัญลักษณ์เหล่านี้ซ้อนทับกันได้อีกเท่าที่ต้องการ แต่จะต้องมีจำนวน "→" เท่ากับ "←" เสมอ ซึ่งอาจจะสามารถเขียนซ้อนทับกันได้

ปุ่มที่กด	Buttom menu
กด Pick ปกติ	BUTTONS 1
Shift + Pick	BUTTONS 2
ctrl + Pick	BUTTONS 3
ctrl + shift + Pick	BUTTONS 4

ตารางที่ 3
สรุปความสัมพันธ์ระหว่างปุ่มที่กดและบัททอนเมนู



รูปที่ 6 แสดงถึงพลูดาวน์เมนูที่ได้จากการคอมพายล์เมนูแล้ว



รูปที่ 7 แสดงเคอร์เซอร์เมนูที่ได้จากการกดปุ่ม shift + Pick

```
***BUTTONS1
;

***BUTTONS2
$p0=test1 $P0=*

***BUTTONS3
$P0=test2 $P0=*

***POP0
[default]

**TEST1
[Shift+Pick]
[Shift+Pick #1]
[Shift+Pick #2]
[Shift+Pick #3]
[Shift+Pick #4]
[Shift+Pick #5]

**TEST2
[Ctrl+Pick]
[Ctrl+Pick #1]
[Ctrl+Pick #2]
[Ctrl+Pick #3]
[Ctrl+Pick #4]
[Ctrl+Pick #5]

***SCREEN
[mymenu]

***POP1
[POPUP]
[Item1]
[->Item2]
[Item21]
[Item22]
[Item23]
[Item24]
[->Item25]
[Item231]
[Item232]
[Item233]
[Item234]
[<-<-Item235]
[Item3]
[Item4]
[Item5]
```

รูปที่ 8 แสดงซอร์สโปรแกรมที่เป็นเมนูไฟล์

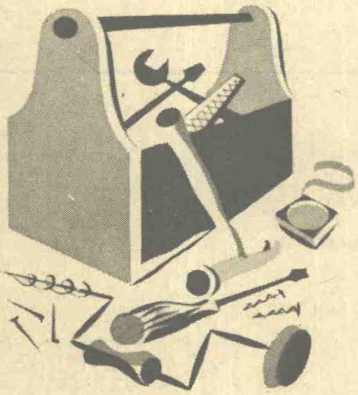
ลองทำเสียที

ที่ผ่านมาเป็นการชี้แจงทฤษฎีพอคร่าว ๆ ซึ่งความจริงยังมีมากกว่านี้ก็คงต้องไปศึกษาเพิ่มเติมเอาเอง แต่ที่ยังไม่ได้กล่าวถึงส่วนอื่น ๆ ก็ไม่ค่อยจะได้ใช้เท่าไรนัก ต่อไปจะเป็นการลงมือปฏิบัติกันจริงเสียที ผมอยากจะให้ท่านพิจารณาในแต่ละบรรทัดให้เข้าใจก่อน จากนั้นท่านลงมือเปลี่ยนแปลงตามความเข้าใจของท่านเพื่อเพิ่มประสบการณ์ไปในตัว แต่อย่าลืมสำรองข้อมูลก่อนด้วยนะครับ ผมจะสร้างพลูดาวน์เมนูให้ซ้อนกัน 3 ชิ้น เฉพาะใน Popl เท่านั้น จะแบ่งออกเป็น ITEM อย่างชัดเจน แต่จะยังไม่ได้ใส่มาโครลงไป ดังในรูปที่ 6

ส่วนของเคอร์เซอร์เมนูก็ให้แสดงเคอร์เซอร์เมนู 2 แบบ คือใช้ปุ่ม shift + Pick แสดงบัททอนที่ 2 และ Ctrl + Pick แสดงบัททอนที่ 3 ในรูปที่ 7 จะเป็นการแสดงบัททอนที่ 2

ส่วนซอร์สโปรแกรมก็จะแสดงดังในรูปที่ 8 แต่ท่านจะต้องจัดเก็บลงในไฟล์ที่มีนามสกุล .MNU เท่านั้น แล้วพิมพ์คำสั่ง

```
Command: MENU
```



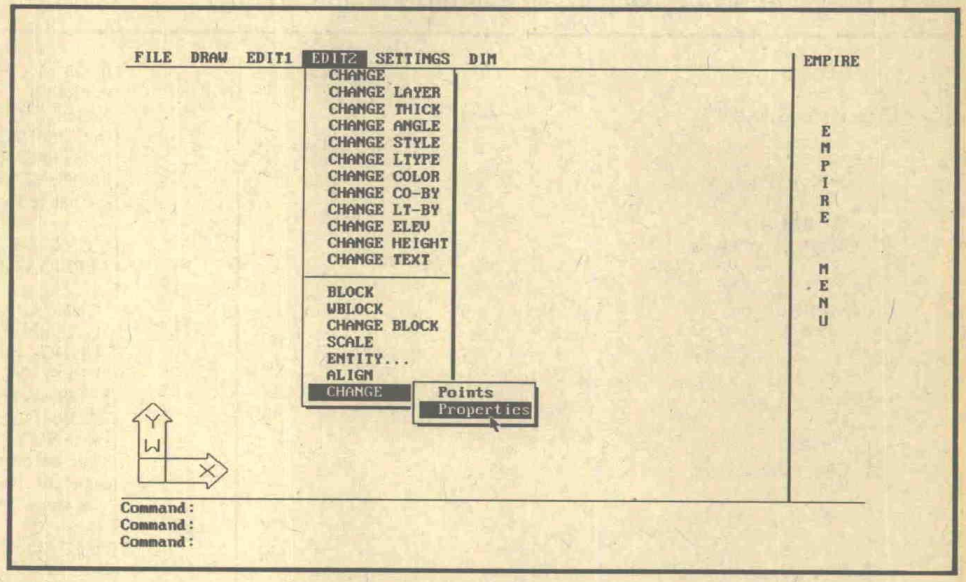
ก็จะเกิดไดอะล็อกบ็อกซ์ขึ้นมาให้ไปเลือก Pick ซึ่งที่ท่านจัดเก็บก็คือเป็นอันใช้ได้ครับ

ประยุกต์ใช้งานจริง

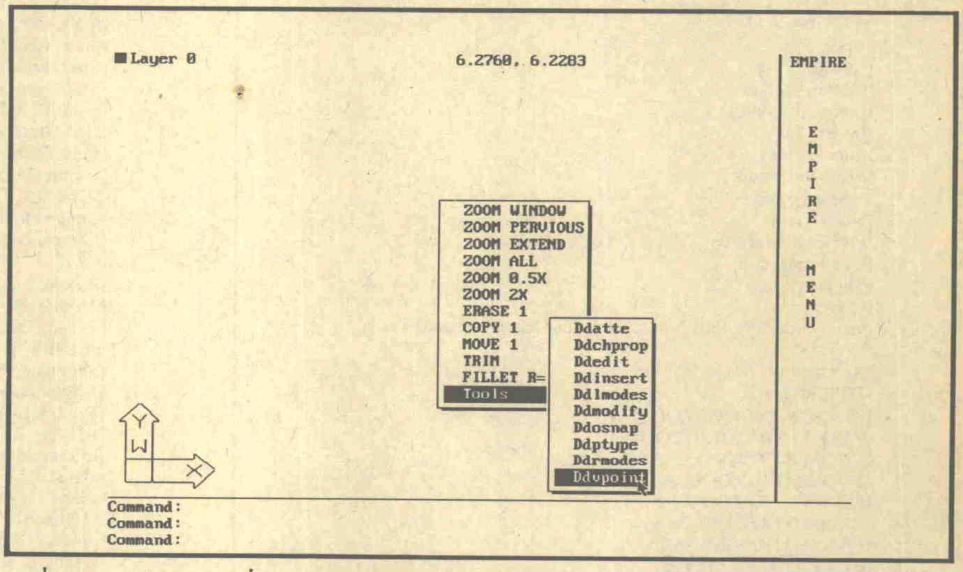
ถึงคราวที่จะต้องใช้งานจริงกันบ้างละ ผมสร้างเป็น 6 Popup แถมด้วยเคอร์เซอร์เมนูอีก 1 ตัว ดังนั้นในเมนูนี้ก็ยังมีเนื้อที่ว่างสร้าง Popup อีก 10 ตัว และเคอร์เซอร์เมนูอีก 2 ตัว ก็คงต้องเป็นฝีมือของท่านบ้างแล้วละครับที่จะต้องนำไปพัฒนากันต่ออีก ซึ่งเมนูนี้เป็นส่วนที่ตัดมาจากเมนูของผมที่ได้ใช้งานจริง ๆ อยู่ในขณะนี้ ส่วนจะถูกใจหรือเปล่าก็คงเป็นหน้าที่ของท่านที่จะนำไปปรุงแต่งเอาเอง ซึ่งหน้าตาของเมนูจะดังเช่นในรูปที่ 9 และรูปที่ 10 ต่อไปจะเป็นซอร์สโปรแกรมของเมนูนี้ (ดูรูปที่ 11)

นำไปพัฒนาต่อ

ท้ายสุดก็คงเป็นเรื่องราวของการนำไปประยุกต์หรือพัฒนากันต่อ ซึ่งผมขอมอบภาระกิจนี้ให้แก่ท่านในการไปปรุงแต่ง และถ้าหากใครขี้เกียจอีดิตก็ติดต่อกับทางวารสารไมโครฯ นะครับ สำหรับฉบับหน้าก็เป็นการกล่าวถึง AME (Advanced Modeling Extension) พ่อมดแห่ง 3D ที่ท่านผู้อ่านเรียกร้องกันมาเหลือเกิน อย่าพลาดนะครับ



รูปที่ 9 แสดงพลูดาวน์เมนูที่สามารถนำไปใช้งานจริงได้



รูปที่ 10 เคอร์เซอร์เมนูที่เป็นการประยุกต์ใช้งานจริง (ปุ่ม ctrl + Pick)

รูปที่ 11 แสดงซอร์สโปรแกรมของเมนูที่ประกอบด้วยมาโครโดยสมบูรณ์แล้ว

```
***Comment
<< The EMPIRE Utilities V1.01 >> developed by VITTAYAR S.
***BUTTONS1
;
$p0=*
^c^c

***BUTTONS2
$p0=P0A $P0=*

***BUTTONS3
$P0=CURSOR $P0=*
```

รูปที่ 11 แสดงซอร์สโปรแกรมของเมนูที่ประกอบด้วยมาโครโดยสมบูรณ์แล้ว (ต่อ)

```
***AUX1
;
$p0=P0A $P0=*

***AUX2
$p0=P0A $P0=*

***SCREEN
[ EMPIRE ]^c^c$s=x

**x
[ EMPIRE ]^c^c



[ E ]
[ M ]
[ P ]
[ I ]
[ R ]
[ E ]

[ ]
[ M ]
[ E ]
[ N ]
[ U ]
[ ]

***POP0
**p0a
[Osnap]
[Center]_center
[Endpoint]_endp
[Insert]_ins
[Intersection]_int
[Midpoint]_mid
[Nearest]_nea
[Node]_nod
[Perpendicular]_per
[Quadrant]_qua
[Tangent]_tan
[None]_non
[Calculator]^c^c^p(if (not c:calc) (load"calc"))^p calc

**CURSOR
[UTILITIES]
[ZOOM WINDOW]'ZOOM W
[ZOOM PERVIOUS]'ZOOM P
[ZOOM EXTEND]'ZOOM E
[ZOOM ALL]'ZOOM A
[ZOOM 0.5X]'ZOOM 0.5X
[ZOOM 2X]'ZOOM 2X
[ERASE 1]^C^CERASE;\;
[COPY 1]^C^CCOPY;\;
[MOVE 1]^C^CMOVE;\;
[TRIM]^C^CTRIM
[FILLET R=0]^C^CFILLET;R;0;FILLET
[->Tools]
[Ddatte ]ddatte
[Ddchprop]ddchprop

[Ddedit ]ddedit
[Ddinsert]ddinsert
[Ddlmodes]ddlmodes
[Ddmodify]ddmodify
[Ddosnap ]ddosnap
[Ddptype ]ddptype
[Ddrmodes]ddrmodes
[<-Ddvpoint]ddvpoint

***POP1
[FILE]
[NEW...]^C^C_new
[OPEN...]^C^C_open
[SAVE...]^C^C_qsave
[SAVE AS...]^C^C_saveas
[PLOT...]^C^C_plot
[->XREF]
[Attach...]^C^C_xref;_attach;~
[Detach]^C^C_xref;_detach
[Reload]^C^C_xref;_reload
[List]^C^C_xref;?;*;
[<-Change Path]^C^C_xref;_path
[--]
[UTILITIES...]'_files
[APPLICATIONS...]^C^Cappload
[--]
[ CANCEL ]^C^C^c^p
[ U-TURN ]^C^CU
[ UNDO ]^C^cUNDO
[ REDO ]^C^CREDO
[REDRAW]'REDRAW
[REGEN]^C^CREGEN
[->MENU]
[Type It]^c^cmenu
[Standard .]^c^cmenu ACAD
[<- Empire ]^c^cmenu empire
[--]
[EXIT AutoCAD]^C^C_quit

***POP2
[DRAW]
[->ARC]
 [3-point]^C^C_arc
 [Start, Cen, End]^C^C_arc;\_c
 [Start, Cen, Angle]^C^C_arc;\_c;\_a
 [Start, Cen, Length]^C^C_arc;\_c;\_l
 [Start, End, Angle]^C^C_arc;\_e;\_a
 [Start, End, Radius]^C^C_arc;\_e;\_r
 [Start, End, Dir]^C^C_arc;\_e;\_d
 [Cen, Start, End]^C^C_arc;_c
 [Cen, Start, Angle]^C^C_arc;_c;\\_a
 [<-Cen, Start, Length]^C^C_arc;_c;\\_l
[->CIRCLE]
 [Center, Radius]^C^C_circle
 [Center, Diameter]^C^C_circle;\_diameter
 [--]
 [2-Point]^C^C_circle;_2p
 [3-Point]^C^C_circle;_3p
 [<-Tan, Tan, Radius]^C^C_circle;_ttr
[->LINE]
 [Segments]^C^C_line
 [1 Segment]^C^Cline;\\;
 [Double Lines]^C^Cdline
 [--]
 [<-Sketch]^C^C_sketch
[->INSERT]
[Insert...]^C^Cddinsert
[<-Minsert]*^C^Csetvar attdia 1 minsert
[->POLYGON]
 [Edge]^C^C_polygon;\_edge;
 [--]
 [Circumscribed]^C^C_polygon;\\_circumscribed;
 [<-Inscribed]^C^C_polygon;\\_inscribed;
 [->ATTRIBUTES]
  [Define...]^C^Cddattdef

  [Edit...]^C^C_ddatte
  [<-Extract...]^C^Cddattext
[->INQUIRE]
 [List]^C^C_list
 [Status]'_status
 [--]
 [Area]^C^C_area
 [Distance]^C^C'_dist
 [<-ID Point]'_id
[--]
[DTEXT]^c^Cdtext
[TRACE]*^C^C$S=X $s=trace trace
[HATCH...]^C^Cbhatch
[PDMODE]^c^cSetvar;pdmode;\;
[OFFSET]*^C^C$S=X $s=Offset Offset;\\
[PLINE ]*^C^C$S=X $s=pline pline
```

รูปที่ 11 แสดงซอร์สโปรแกรมของเมนูที่ประกอบด้วยมาโครโดยสมบูรณ์แล้ว (ต่อ)

```
[POINT]*^C^C$S=X $s=Point Point;\;
[RECTANGLE]^C^Crectang
[CALCULATOR]^c^c^p(if (not c:calc) (load"calc"))^p calc
[--]
[HELP!...]'?

***POP3
[EDIT1]
[Array ]*^c^c$S=X $s=array array
[->BREAK]
[Break2]*^c^c$S=X $s=break break
[<-Break1]*^c^c$S=X $s=Break Break;\F;\@;
[->COPY]
[Copy]$M=$(if,$(eq,$(substr,$(getvar,cmdnames),1,4),GRIP),_copy,^C^C_copy)
[<-Multiple]^c^c_Select \ ^c^c_copy P; m
[DIVIDE]*^c^c$S=X $s=divide Divide
[->ERASE]
 [Select]^C^C_erase si auto
 [Single]^C^C_erase;_single;;
 [Window]^C^C_erase;_w;
 [Crossing]^C^C_erase;_c;
 [Last]^C^C_erase;_last;;
  [All]^C^C_erase;_ALL;;
 [--]
 [<-Oops!]^C^C_oops
[EXPLODE  .]*^c^c$S=X $s=Explode Explode
[EXTEND]*^C^C$S=X $s=extend extend auto
[FILLET]*^c^c$S=X $s=Fillet Fillet
[MOVE]*^C^Cmove si auto
[MIRROR]$M=$(if,$(eq,$(substr,$(getvar,cmdnames),1,4),GRIP),_mirror,^C^C_mirror)
[MEASURE]*^c^c$S=X $s=Measure Measure
[OFFSET]*^c^c$S=X $s=Offset Offset;\\
[PEDIT]*^C^C$S=X $s=pedit pedit
[ROTATE]*^c^c$S=X $s=Rotate Rotate crossing \\;\\;;
[STRETCH]*^C^C$S=X $s=stretch stretch crossing \\;\\;;
[SCALE]*^c^c$S=X $s=Scale Scale crossing \\;\\;;
[TRIM]*^C^Ctrim si auto
[DONUT]^C^C_donut
[--]
[CONTOUR]*^C^C$s=pline pline;\a;s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;+
;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;\\s;

***POP4
[EDIT2]
[CHANGE]*^c^c$S=X $s=change change
[CHANGE LAYER]*^c^c$S=X $s=Change Change;\;p;La;\;
[CHANGE THICK]*^c^c$S=x $s=Change Change;\;p;Th;\;
[CHANGE ANGLE]*^C^C$S=X $s=Change Change;\;;;;;\;;
[CHANGE STYLE]*^C^c$s=X $s=Change Change;\;;;\;;
[CHANGE LTYPE]*^c^c$S=X $s=Change Change;\;p;Lt;\;
[CHANGE COLOR]*^c^c$S=X $s=Change Change;\;p;Col;\;
[CHANGE CO-BY]*^c^c$s=X $s=Change Change;\;p;Col;bylayer;;\;
[CHANGE LT-BY]*^c^c$S=X $S=Change change;\;p;lt;bylayer;;\;\;
[CHANGE ELEV]*^c^c$S=X $s=Change Change;\;p;El;\;
[CHANGE HEIGHT]*^C^C$S=X $s=Change Change;\;;;;\;;
[CHANGE TEXT]*^C^C$S=X $s=Change Change;\;;;;;;
[--]
[BLOCK]^c^cblock;\;;\w;\\;
[WBLOCK]^c^cwblock;\;\w;\\;

[CHANGE BLOCK]^c^c^p(if (not c:CHBLOCK) (load"CHBLOCK"))^p CHBLOCK
[SCALE]$M=$(if,$(eq,$(substr,$(getvar,cmdnames),1,4),GRIP),_scale,^C^C_scale)
[ENTITY...]^C^Cddmodify
[ALIGN]^C^Calign
[->CHANGE]
  [Points]^C^C_change
  [<-Properties]^C^Cddchprop

***POP5
[SETTINGS]
[AXIS]$S=X $S=AXIS ^C^CAXIS
[BLIPS]$S=X $S=BLIPMODE ^C^CBLIPMODE
[DRAGMODE]$S=X $S=DRAGMODE ^C^CDRAGMODE
[LTSCALE]^C^CLTSCALE
[SETVARIABLES]$S=X $S=SETVAR ^C^CSETVAR
[SNAP]$S=X $S=SNAP ^C^CSNAP
[STYLE FONTS]$S=X $S=STYLE ^C^CSTYLE
[TABLETS]$S=X $S=TABLET ^C^CTABLET
[DRAWING AIDE...]'_ddrmodes
[OBJECT SNAP...]'ddosnap
[--]
[ENTITY MODES...]'_ddemodes
[POINT STYLE...]'ddptype
[--]
[DIMENSION STYLE...]^C^Cddim
[UNITS CONTROL...]'ddunits
[--]
[SELECT SETTINGS...]'ddselect
[GRIPS...]'ddgrips
[--]
[DRAWING LIMITS]'_limits;;

***POP6
[DIM]
[DIM ASSOCIATE]^c^cDim;dimaso;
[DIM SCALE]^c^cDim;Dimscale;
[DIM VERTICAL]*^c^cDim;ver;\\\\;
[DIM DIAMETER]*^c^cDIM;DIA;\\\;
[DIM RADIOUS]*^C^CDIM;RAD;\\\;
[DIM HORIZONTAL]^c^cDim;hor;\\\\;
[DIM CONTINUES]^c^cDim;con;\\;
[DIM ANGULAR]^c^c^PDim;angular;
[DIM LEADER]^c^c^PDim;leader;\\\;\;
[DIM STATUS]^c^c^PDim;status;
[DIM STYLE]^C^c^PDIM;STYLE
[EXIT]^C^C^C^C
[--]
[SAVE ]^C^C^PDIM;SAVE
[RESTORE]^C^C^PDIM;RESTORE
[MOVETEXT]^C^C^PDIM;TEDIT
[HOMETEXT]^C^CDIM;$S=X $S=HOMETEXT HOMETEXT
[UPDATE TEXT]^C^CDIM;$S=X $S=UPDATE UPDATE
[NEWTEXT]^C^CDIM;$S=X $S=NEWTEXT NEWTEXT
[MARK CENTER]*^C^CDIM;CENTER;\;\;
```